

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

**WEB OF SCIENCE™**

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Role of Finite Element Analysis in Designing Multi-axes Positioning for Robotic Manipulators

T.T. Mon, F.R. Mohd Romlay and M.N. Tamin
*Universiti Malaysia Pahang, Universiti Teknologi Malaysia
 Malaysia*

1. Introduction

Simulation of robot manipulator in Matlab/Simulink or any other mechanism simulator is very common for robot design. However, all these approaches are mainly concerned with design configuration having little analysis meaning that the robot model is formed by linking the kinematics and solid description, and simulated for alternative configuration of movements (Cleery & Mathur, 2008). Indeed comprehensive design should have analysis at different computational levels. Finite element method (FEM) has been a major tool to develop a computational model in various fields of studies because of its modelling and simulation capability close to reality. Subsequently, modelling and analysis with FEM has become the most convenient way to economically design and analyze real world problems, either in static or dynamic. As a result, huge amount of reports on this topic can be found in the literature (Mackerle, 1999). Unfortunately however, this technique has not comprehensively applied in designing a robot while choosing the best components for the design is as important as having good performance and no environmental impact of the machines over its lifetime. Building block of a robot manipulator is electromechanical system in which mechanical systems are controlled by sophisticated electric motor drives. Since energy saving everywhere is a major challenge now and in future, getting electromechanical design right will significantly contribute to energy saving.

This chapter is dedicated to the application of Finite Element Method (FEM) in designing multi-axes positioning for robot manipulators. Computational model that can predict physical behaviour of dynamic robot manipulators constructed using FE codes is presented, and this is major contribution of the chapter. FEM tools necessary for modelling and analysis of multi-axes positioning are presented in large part. Rather than a FEM discourse, FEM is presented by highlighting mathematics behind and an application example as they relates to practical robotic manipulation. It is, however, assumed that the reader has acquired some basic knowledge of FEM consistent with the expected level of mathematics. Hence, the chapter is organized as follows.

In the early part of the chapter, the important terminologies used in robotics are defined in the background. The material is presented using a number of examples as evidenced in the

published reports. Then the chapter will go to mathematics behind finite element modelling and analysis of kinematics of a structure in 3D space as robotics involves tracking moving objects in 3D space. This will also include mathematical tools essential for the study of robotics, particularly matrix transforms, mathematical models of robot manipulators, direct kinematic equations, inverse kinematic technique and Jacobian matrix needed to control position and motion of a robot manipulator. More emphasis will be on how these mathematical tools can be linked to and incorporated into FEM to carry out design analysis of robot structure.

The rest of the chapter will present application of FEM in practical robot design, detailed development of FE model computable for multi-axes positioning using a particular FE code (ALGOR, 2008), and useful results predicted by the computational model. The chapter will be closed by concluding remarks to choice of FE codes and its impact on the computational model and finally the usefulness of computational model.

2. Background in Robotics

Multi-axis positioning meant here is different movements of a point, or a structure in different directions. This term is drawn from the term usually come with computer numerical controlled machines just as 3-axis, 5-axis and so on, where the 3-axis machine, for instance, implies that it can make a maximum of three different positioning of the controlled elements. Each axis is alternatively referred to as degree of freedom (DOF) that is something to do with motion in a system or a structure. Since the term 'axis' is adopted to represent an element that creates motion, 3-axis positioning means three DOF's, for example (Rahman, 2004). In relation to these definitions, one manipulator of a robot can represent one axis or DOF as the manipulator is the robot's arm, a movable mechanical unit comprising of segments or links jointed together with axes capable of motion in various directions allowing the robot to perform tasks. Typically, the body, arm and wrist are components of manipulators. Movements between the various components of the manipulator are provided by series of joints.

The points that a manipulator bends, slides or rotates are called joints or position axes. Position axes are also called the world coordinates. The world coordinate system is usually identified as being a fixed location within the manipulator that serves as an absolute frame of reference.

In general, the manipulator's motion can be divided into two categories: translation and rotation. Although one can further categorize it in specific term such as a pitch (up-and-down motion); a yaw (side-to-side motion); and a roll (rotating motion), any of these is fall into either translation or rotation. The individual joint motion associated with either of these two categories is referred to degree of freedom. Subsequently, one degree of freedom is equal to one axis. The industrial robots are typically equipped with 4-6 axes.

The power supply provides the energy required for a robot to be operated. Electricity is the most common source of power and is used extensively with industrial robots. Payload is the weight that the robot is designed to lift, hold, and position repeatedly with the same accuracy. Hence, the power supply has direct relation to the payload rating of a robot.

Among the important dynamic properties of a robot that properly regulates its motion are: stability, control resolution, spatial resolution, accuracy, repeatability and compliance. To take these factors into account in the design of a robot is a complex issue. Lack of stability

occurs very often due to wear of manipulator components, movement longer than the intended, longer time to reach and overshooting of position.

Control resolution is all about position control. It is a function of the design of robot control system and specifies the smallest increment of motion by which the system can divide its working space. It is the smallest incremental change in position that its control system can measure. In other words, it is the controller's ability to divide the total range of movements for the particular joint into individual increments that can be addressed in the controller. This depends on the bit storage capacity in the control memory. For example, a robot with 8 bits of storage can divide the range into 256 discrete positions. The control resolution would be also defined as the total motion range divided by the number of increments. For example, a robot has one sliding joint with a full range of 1.0 m. The robot control memory has 12-bit storage capacity. The control resolution for this axis of motion is 0.244mm. The spatial resolution of a robot is the smallest increment of movement into which the robot can divide its work volume.

Mechanical inaccuracies in the robot's links and joint components and its feedback measurements system (if it is a servo-controlled robot) constitute the other factor that contributes to spatial resolution. Mechanical inaccuracies come from elastic deflection in the structural members, gear backlash, stretching of pulley cords and other imperfections in the mechanical system. These inaccuracies tend to be worse for large robot simply because the errors are magnified by the large components. The spatial resolution is degraded by these mechanical inaccuracies.

Rigidity of the structure also affects the repeatability of the robot. Compliance is a quality that gives a manipulator of a robot the ability to tolerate misalignment of mating parts. It is essential for assembly of close-fitting parts. In an electric manipulator, the motors generally connect to mechanical coupling. The sticking and sliding friction in such a coupling can cause a strange effect on the compliance, in particular, being back-drivable.

An Off-line programming system includes a spatial representation of solids and their graphical interpretation, automatic collision detection, incorporation of kinematic, path planning and dynamic simulation and concurrent programming. The off-line programming will grow more in the future because of graphical computer simulation used to validate program development. It is important both as aids in programming industrial automation and as platforms for robotic research (Billingsley, 1985; Keramas, 1999; Angeles, 2003).

3. Mathematical Foundation

Forward and reverse kinematics methods are the principal mathematics behind typical modeling, computation and analysis of robot manipulators. Since the latter is deduced from the former, broader review will focus on some related mathematics of the former. Forward kinematic equation relates a pose element to the joint variables. The pose matrix is computed from the joint variables. The position and orientation of end-manipulator (also-called the end-effector, the last joint that directly touches and handles the object) is computed from all joint variables. The position and orientation of the end manipulator are computed from a set of joint variable values which are already known or specified. The computation follows the arrow directions starting from joint 1 as illustrated in Fig. 1. It should be noted that in kinematics analysis, the manipulators are assumed to be rigid.

3.1 Defining the location of an object in space

As a general case, the object location in 3D space is considered. A matrix representation is widely used to represent the object location as it is convenient and easy to handle especially when the location is changed. Two parameters are needed to define the object location: position and orientation. Basically, a homogenous vector 'v' is represented as

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} \quad (1)$$

if v is a free vector or

$$\mathbf{v} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

if v represents the position of a particular point in the usual coordinates system (x, y, and z). This coordinates system is referred to a frame. These frames are used to track an object location in space. As shown in Fig. 2, the frame F_0 is attached to a fixed point while another F_A to an object. The object position is described by the vector p_A of the origin A of the frame F_A . The orientation of the object is given by the homogeneous vectors of each unit vectors x_A , y_A and z_A of F_A with respect to F_0 .

Then the object location is mathematically represented by a post matrix 'P' as:

$$\mathbf{P} = [x_A \ y_A \ z_A \ p_A] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where the matrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (4)$$

and

$$\mathbf{p}_A = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (5)$$

formed by the coordinates of the vectors x_A , y_A and z_A is a rotation matrix that holds the orientation of the object while the \mathbf{p}_A holds the position of the object. In a compact form, the pose matrix can be written as:

$$\mathbf{P}=\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$$

(6)

where $\mathbf{0}$ refers to a 1×3 vector of zeros.

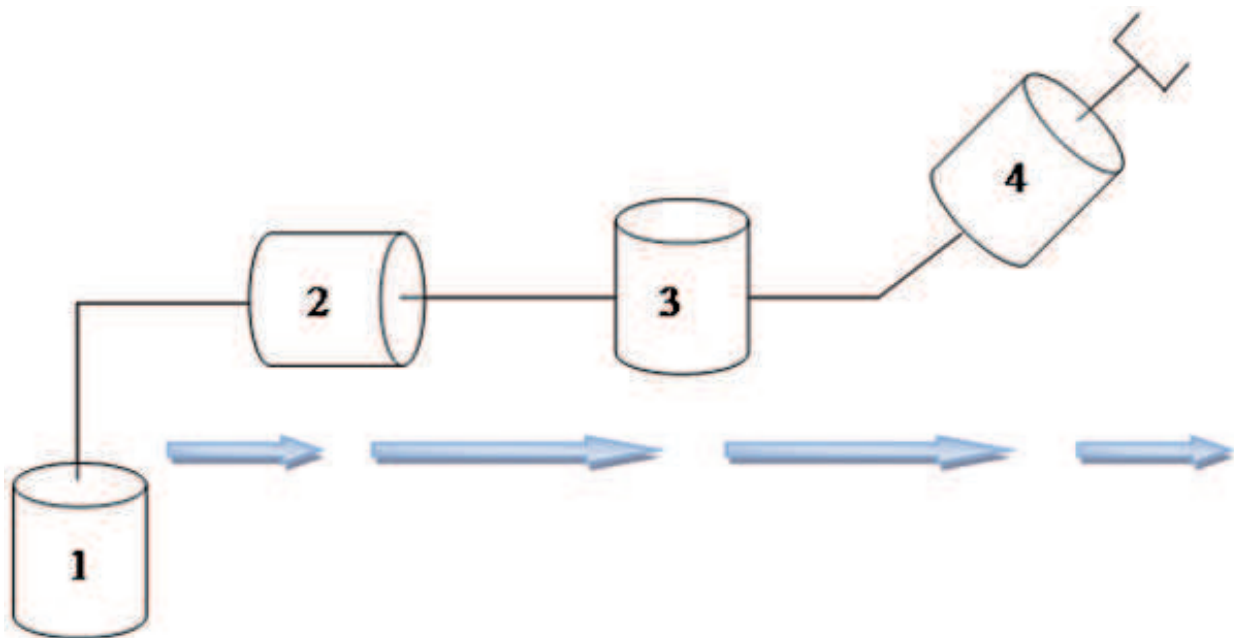


Fig. 1. Forward kinematics

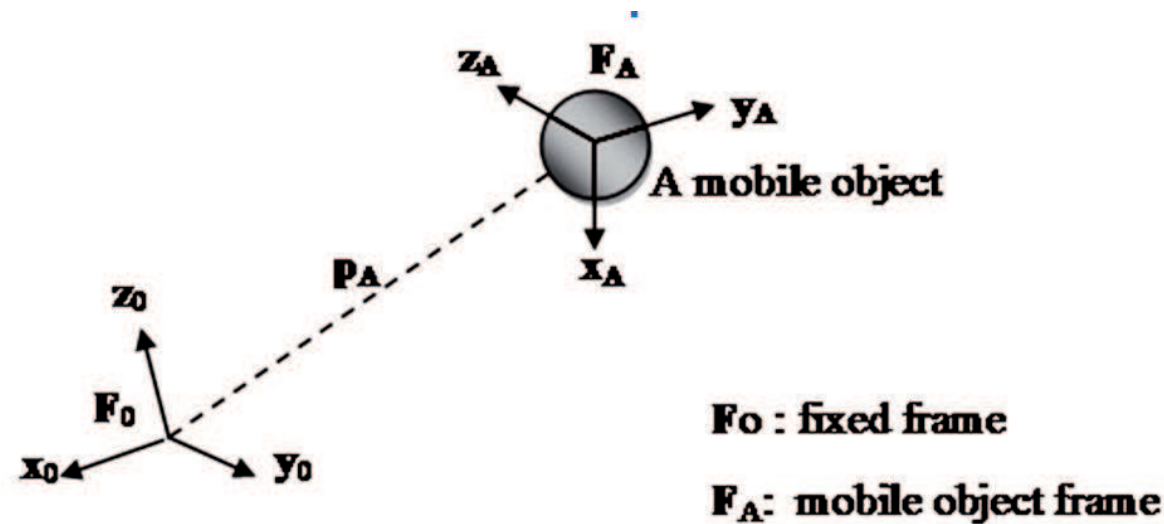


Fig. 2. The object with respect to frames.

When a point, say Q given by its coordinate vector ${}^Aq = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ with respect to frame F_A , is transformed to the frame F_0 , the transformed vector say 0q can be expressed as:

$${}^o\mathbf{q} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = R^A \mathbf{q} + \mathbf{p} \quad (7)$$

In compact form

$${}^o\mathbf{q} = {}^oT_A^A \mathbf{q} \quad (8)$$

where ${}^oT_A^A = P$

Similarly in alternative representation of frame transforms, simple space translation and rotation can be conveniently represented as a compact-form matrix as:

$${}^T\mathbf{q} = T_u \mathbf{q} \quad (9)$$

$${}^R\mathbf{q} = R_{v,\theta} \mathbf{q} \quad (10)$$

where $\mathbf{q} = \begin{bmatrix} x_q \\ y_q \\ z_q \end{bmatrix}$ is coordinates vector of a point Q, T_u is a translation vector, $R_{v,\theta}$ is a rotation

vector, ${}^T\mathbf{q}$ is coordinates vector of a point Q' where Q is translated by T_u and ${}^R\mathbf{q}$ coordinates vector of a point Q' where Q is rotated around \mathbf{v} by an angle θ .

Furthermore, the translations and rotations along the reference axes, called canonical translations/rotations, have the homogeneous matrix of the following form respectively as:

$$T_x(d) = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$T_y(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$T_z(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix} \quad (14)$$

$$R_x(\beta) = \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \quad (15)$$

$$R_z(\gamma) = \begin{bmatrix} c_\gamma & -s_\gamma & 0 \\ s_\gamma & c_\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Where $T_x(d)$, $T_y(d)$, and $T_z(d)$ are translations of a distance, 'd' in x, y and z directions respectively, and $R_x(\alpha)$, $R_y(\beta)$, and $R_z(\gamma)$ are rotations about the respective axis by the respective angle with $c_{(\cdot)} = \cos(\cdot)$ and $s_{(\cdot)} = \sin(\cdot)$. These can be expanded to arbitrary transformations. Refer to Manseur (2006) for details.

3.2 DH parameters

In the conventional analysis of motion of robot manipulators, the Denavit-Hartenberg (DH) modelling technique is commonly used as a standard technique. Reference frames are assigned to each link based on DH parameters, starting from the fixed link all the way to the last link. The DH model is obtained by describing each link frame with respect to the preceding link frame. The original representation of one frame with respect to another using pose matrix requires a minimum of six parameters. The DH modelling technique reduces these parameters to four, routinely noted as:

- di, the link offset,
- ai, the link length,
- θ_i , the link angle and
- α_i , the link twist as illustrated in Fig. 3.

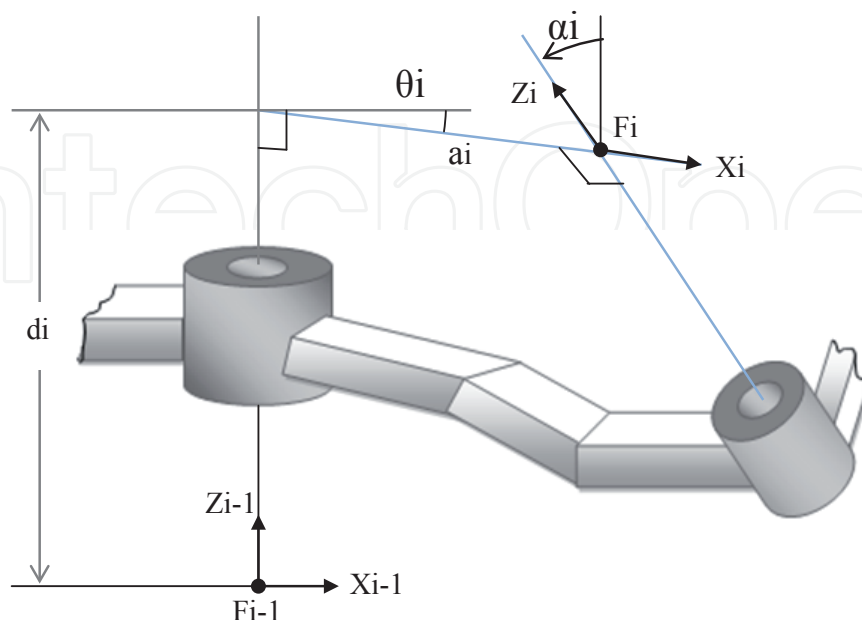


Fig. 3. Illustration of DH parameters

Referring to Fig. 4, in relation to the link frames and DH parameters, homogenous frame transform from the frame (i) to (i-1) can be generally written as

$${}^{i-1}A_i = T(z, d_i) R(z, \theta_i) T(x, a_i) R(x, \alpha_i) \quad (17)$$

where $T(z, d_i)$ is translation from F_{i-1} to F_d , $R(z, \theta_i)$ the rotation transform from F_d to F_θ , $T(x, a_i)$ the translation transform from F_θ to F_a , $R(x, \alpha_i)$ the rotation transform from F_a to F_i .

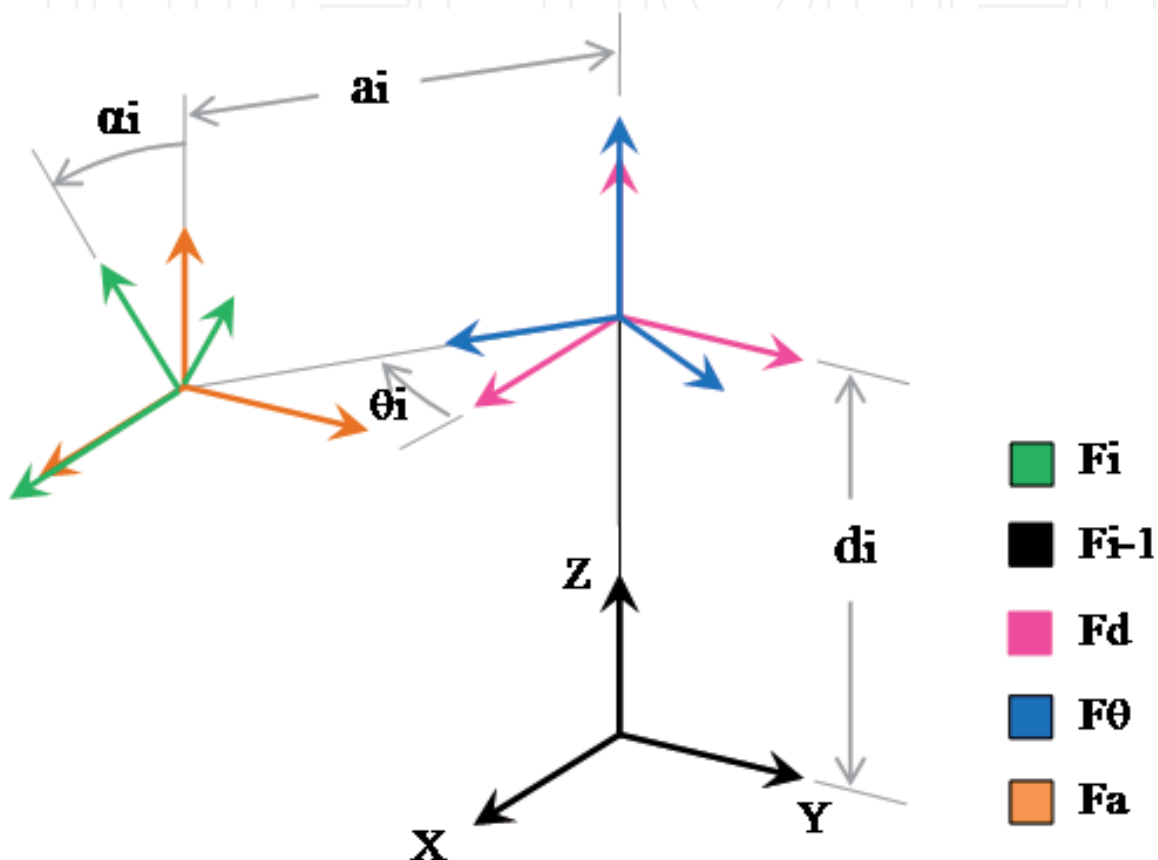


Fig. 4. Frame transformation of the DH parameters.

The matrix ${}^{i-1}A_i$ is then used to express the transform of the end frame to the base frame for 'n-axis' or 'n-joint' robot manipulators. That is

$${}^o u = A_1 A_2 \dots A_{n-1} A_n {}^n u \quad (18)$$

or

$${}^o u = T_n {}^n u \quad (19)$$

where $T_n = A_1 A_2 \dots A_{n-1} A_n = \mathbf{P}$ = pose matrix of the end-effector. This process is called forward kinematic in modelling, computation and analysis of robotic manipulator.

In the reverse kinematics method, as it means, one or more sets of joint variables are computed from the known end-effector pose matrix. As such the direction of computation is

reversed from that of the forward kinematics as illustrated in Fig.5. Nevertheless, the same basic concept of forward kinematics is applied except the fact that the mathematical manipulation becomes different and sophisticated. To be brief, the equations derived from the forward kinematics method are reduced to a set of equations involving as few unknown joint variables as possible using a kinematic function that depends on only certain joint variables. This results simplified set of equation that can be solved analytically. Details can be found elsewhere (Manseur, 2006).

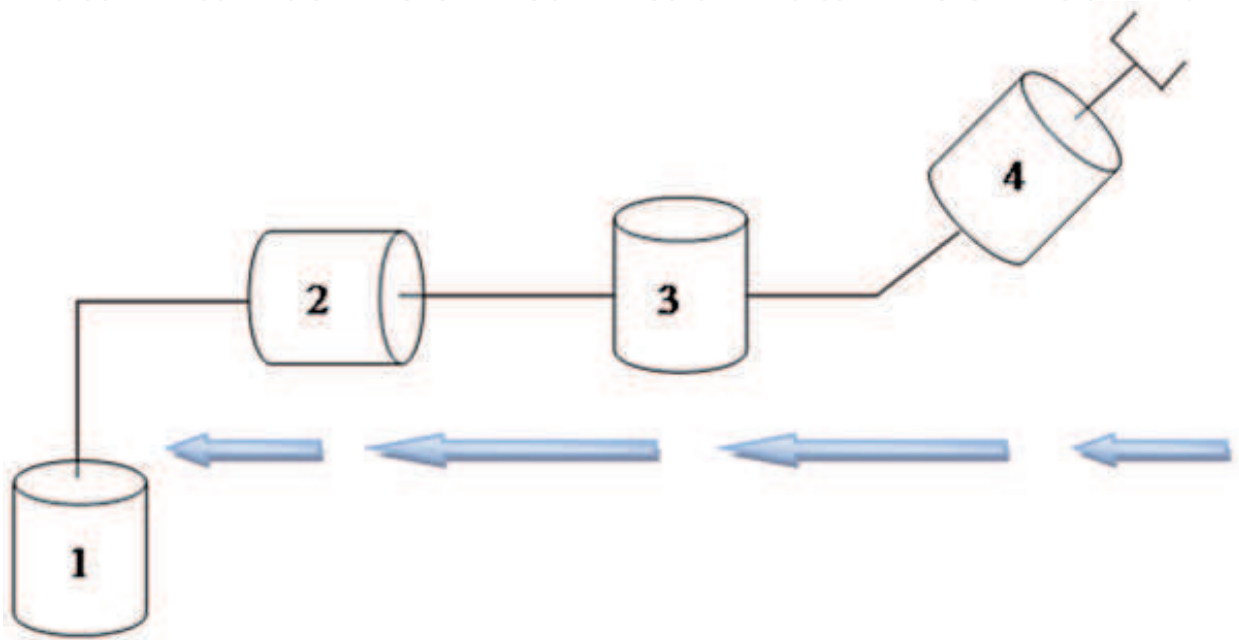


Fig. 5. Reverse kinematics

3.3 Velocity Kinematics

When the time factor is taken into account in the movement of a manipulator, *Jacobian* matrix is used to manipulate the transformation of manipulator. The generalized velocity vector \mathbf{V} of the end-effector with respect to the base frame can be a form of a linear velocity vector \mathbf{v} and an angular velocity vector $\boldsymbol{\omega}$:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \quad (20)$$

where

$$\mathbf{v} = \begin{bmatrix} dp_x/dt \\ dp_y/dt \\ dp_z/dt \end{bmatrix} \quad (21)$$

and

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (22)$$

For the case of n-joint robot, the relationship between the V and the joint velocities can generally be expressed as

$$V = J(q) \dot{q} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix} \dot{q} \quad (23)$$

Where $q = [q_1 \ q_2 \ \dots \ q_n]^T$ is the joint variable and $\dot{q} = [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_n]^T$ is corresponding velocity vector. The matrix $J(q)$ is called the Jacobian matrix. It is the mathematical relationship between joint motion and end-effector motion. It plays an important role in the analysis and control of robotic motion owing to the fact that it establishes a linear relation between velocity vectors in Cartesian space and in joint space. Details of derivation of J and its computation can be found in (Manseur, 2006).

3.4 Typical Robot Simulators

Nowadays softwares are readily available to generate and solve the kinematic equations. BUILD, SKEG and SIMULINK are some of those. Typical robot simulators allow the programming of a manipulator positioning in Cartesian space and relate a Cartesian co-ordinate set to the robot model's joint angles known as the inverse kinematics transformation (Zlajpah, 2008).

Kinematic description is made up of joint rotation and position information. The inverse kinematic links the kinematics and solid descriptions. During the linking process, a sequence of procedure is called to solve a particular position of the robot. For example, the method of solving 6 DOFs robot as shown in Fig. 6 requires the following steps:

- i. From tool position and orientation, calculate wrist position
- ii. Calculate angle of rotation of axis 1
- iii. Calculate angle 2 and 3 from the two link mechanism connecting the shoulder and the wrist
- iv. Calculate the remaining joint angles using spherical trigonometry [1]

Robot simulators do not allow dynamic effects or systematic errors in either trajectory calculation or inverse kinematic processing. The implementation of a full dynamic model would be out of question for a general simulator (e.g. BUILD simulator).

Robot errors can be categorized into two: static error related to end points and dynamic error related to path. The formers result from deviation of the achieved position from the demand position. The latter comes from mismatch between a desired path and the true path. The computation of such information is very appropriate for a solid modelling system in which methods for analyzing space are well developed. However in these simulations, there will be differences between the actual robot performance and the modelled robot performance. The difference will depend on the assumptions made when designing robot model.

A particular robot's physical kinematics will deviate from the idealized model due to manufacturing tolerances and this will lead to a reduction in the accuracy of the robot's

positioning with respect to a Cartesian space. The weight affects inaccuracies due to compliance of robot manipulators (Billingsley, 1985).

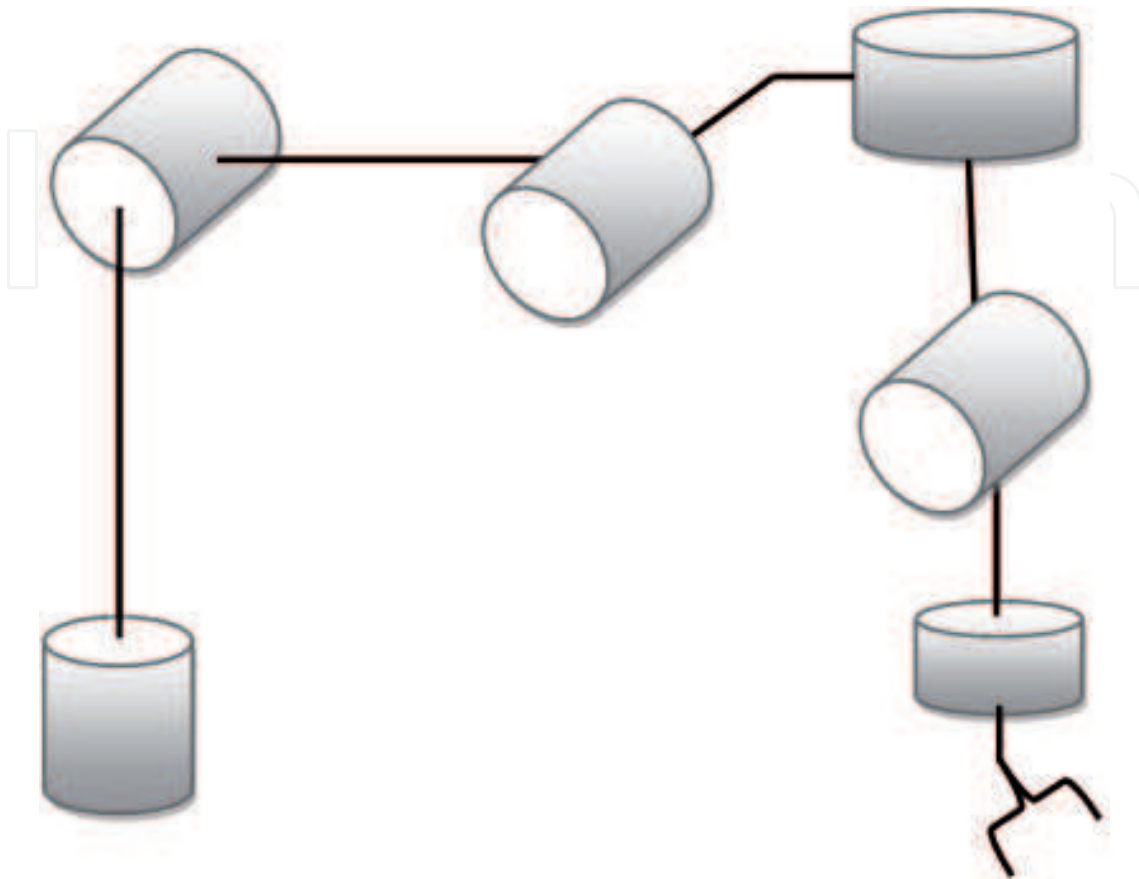


Fig. 6. Robot with 6 DOFs.

To sum up, in conventional modelling technique, the robot manipulators are modelled as a kinematic model. The accuracy of simulated motion in this technique is questionable. Additionally, the model is inadequate for the analysis of the dynamics laws of motion in which inertial parameters, link masses and shapes, and applied forces and torques must be considered. This is where FEM comes into play.

4. Finite Element Method

It is hard to completely define finite element method (FEM) in one sentence or even a paragraph. One may define it as a numerical method for solving engineering problem and physics, or a method to computationally model reality in a mathematical form; either one is acceptable indeed. However, for more complete definition of FEM, the following paragraph may suit it better.

“A continuum is discretized into simple geometric shapes called finite elements; constitutive relations, loading and constraints are defined over these elements; assembly of elements results set of equations; solution of these equations gives the approximate behaviour of the continuum.”

The procedure of using FEM to analyze the system behaviour generally includes:

- Discretize the system and select element type.
- Select displacement function.
- Define strain-displacement and stress-strain relation.
- Derive stiffness matrix and equations.
- Assemble global equations.
- Introduce boundary condition (BC)
- Solve for unknown variables.

Finite element method (FEM) is now considered matured to develop a computational model to predict solution domain accurately without the need of expensive experimental work. Some advantages of using FEM are being economically-viable technique, flexibility to analyze various physical problems, and capability of modelling complex boundary conditions and material behaviour. Most importantly, the aspects of FEM applications are easily adaptable to engineers' needs (Chandrupatla, & Belegundu, 2002; Hutton, 2004).

4.1 Multi-axes positioning with FEM

Multi-axis positioning in FEM can be modelled with manipulation of parameters that can be found under the category of boundary conditions. The concept of DOF here can be illustrated in a simple spring-mass system. Fig. 7(a) shows a system consisting of one spring and one mass where one DOF is imposed on the mass as it displaces due to the weight or applied force. When there is more than one spring or mass, DOF also becomes more than one as shown in Fig. 7(b). In both cases, DOF is in translation. In general structural dynamics, motion can be in translation or rotation defined in standard coordinate axes, x , y or z . These motions are illustrated in Fig. 7(c) where U_x , U_y , and U_z are translations in x , y and z directions while R_x , R_y , and R_z are rotations about x , y and z axes respectively. Hence, in any general-purpose finite element (FE) code, maximum of six DOFs can be defined at a node point for structural analysis.

4.2 Structural analysis with FEM

Basically, computation in FEM involves developing matrix equations for the system and manipulation of these equations. Therefore, one may say that the heart of FEM is rooted in matrix manipulation. For simplest static problem shown in Fig. 8 the system is discretized into two spring elements and three nodes. The FE equation for this system with reference to the coordinate axes can be derived as:

$$\begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} \quad (24)$$

Where k_1 and k_2 are spring constants, U_1 , U_2 , and U_3 are displacements or DOFs at nodes, and F_1 , F_2 , and F_3 are applied forces.

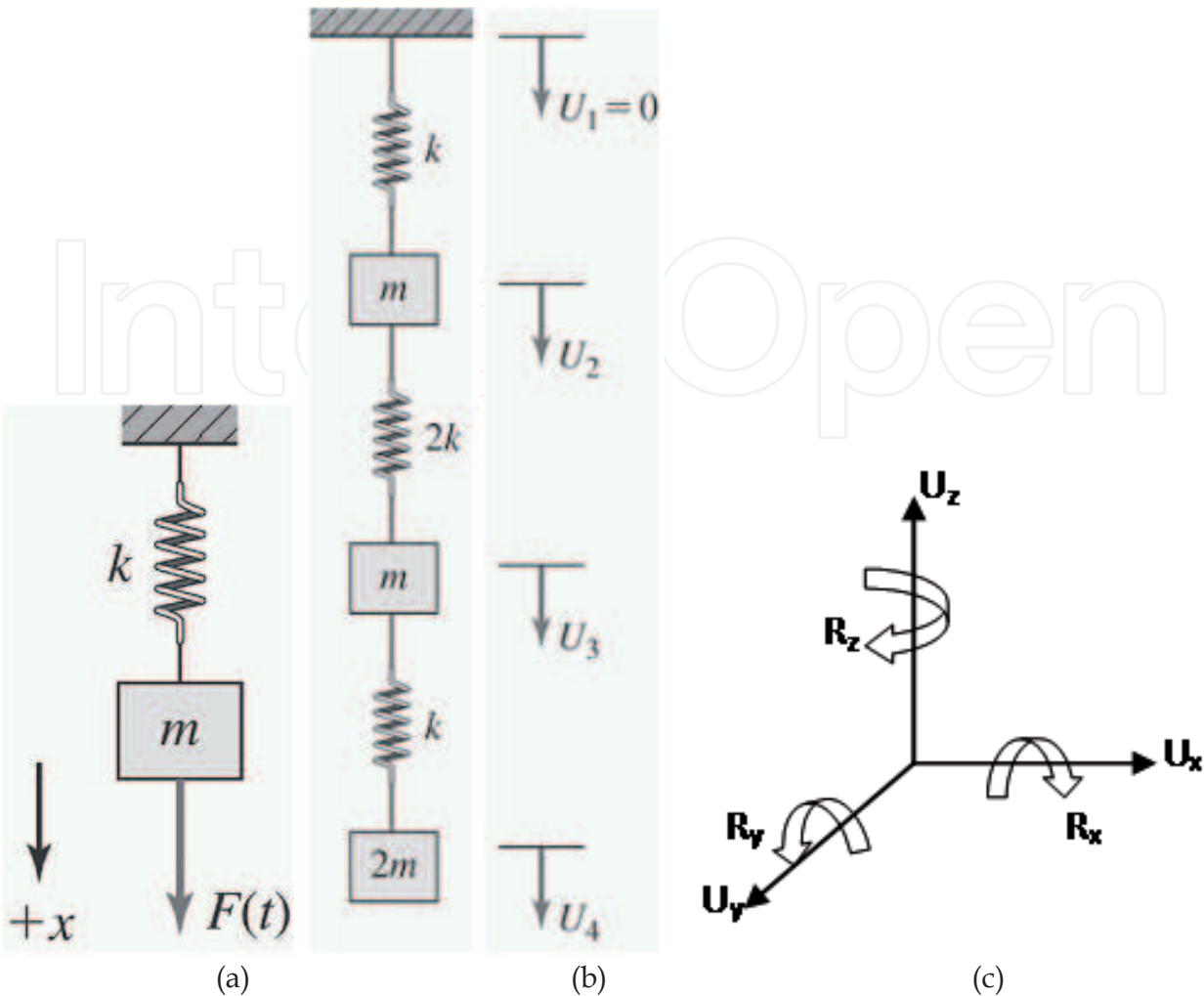


Fig. 7. Motion in standard coordinate axes

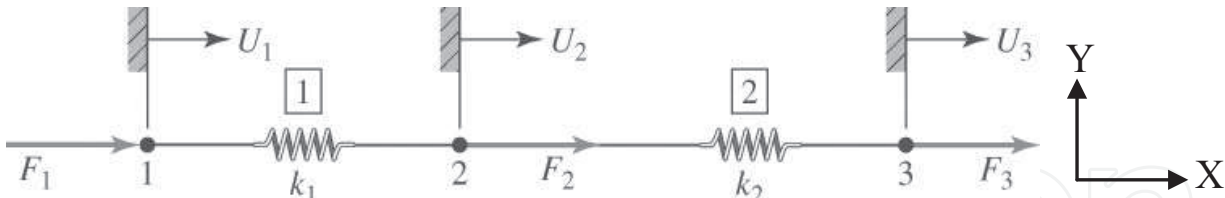


Fig. 8. FE model with spring elements

Detail of derivation procedure is available in (Hutton, 2004). In order to solve for any unknown in equation 24, all possibly-known BC's and forces are imposed. The matrix equations are then solved using an appropriate solver. Thus in FEM, the coordinate axes can be defined up to the user, the kinematics and physics of the system are taken care of in BC and material model respectively, and all necessary computations are performed by the solver. The FE equations are becoming complicated and the more advanced mathematics is required to solve when the system is in dynamics and constructed of complicated geometry in 3D where the element geometry is not so simple as the above example, and also number of elements can go to thousands. For dynamics analysis, finite element formulation is usually based on Lagrangian mechanics which states:

$$L = T \quad (25)$$

where L is Lagrangian term and T is total kinetic energy of the bod. In dynamic problem, mass and acceleration effects come into picture. For 3D body, T can be expressed as:

$$T = \frac{1}{2} \iiint (\dot{u}^2 + \dot{v}^2 + \dot{w}^2) \rho \, dx \, dy \, dz \quad (26)$$

Where ρ is mass density of the body, and \dot{u} , \dot{v} , and \dot{w} are velocity components of differential mass associated with displacements (u, v, w) in the coordinate directions. If computational domain is discretized by

$$u(x, y, z, t) = \sum_{i=1}^M N_i(x, y, z) u_i(t) \quad (27)$$

$$v(x, y, z, t) = \sum_{i=1}^M N_i(x, y, z) v_i(t) \quad (28)$$

$$w(x, y, z, t) = \sum_{i=1}^M N_i(x, y, z) w_i(t) \quad (29)$$

the element kinetic energy can then be expressed as

$$T^{(e)} = \frac{1}{2} \iiint \left(\{\dot{u}\}^T [N]^T [N] \{\dot{u}\} + \{\dot{v}\}^T \{N\}^T [N] \{\dot{v}\} + \{\dot{w}\}^T [N]^T [N] \{\dot{w}\} \right) \rho \, dV^{(e)} \quad (30)$$

In compact form,

$$T^{(e)} = \frac{1}{2} \{\dot{\delta}\}^T [m^{(e)}] \{\dot{q}\} \quad (31)$$

being nodal velocities as

$$\{\dot{q}\} = \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{v}\} \\ \{\dot{w}\} \end{Bmatrix} \quad (32)$$

and the element mass matrix as

$$[m^{(e)}] = \iiint \begin{bmatrix} [N]^T [N] & 0 & 0 \\ 0 & [N]^T [N] & 0 \\ 0 & 0 & [N]^T [N] \end{bmatrix} \rho dV^{(e)} \quad (33)$$

Then based on the Lagrangian approach, the global equations for the FE model of a structure subjected to dynamic loading can be expressed in the form

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, P \quad (34)$$

where P is the total number of DOFs. Solving these equations yields a general system of ordinary equation:

$$[M] \{\ddot{q}\} + [K] \{q\} = \{F\} \quad (35)$$

Finite element solver will solve equation (35) for displacements thereby strains and stresses.

5. Application Example

Application of FEM in modelling and analysis of multi-axis positioning is demonstrated in this section. The real robot manipulators (model: Fanuc M-6iB) shown in Fig. 9 is used as a reference for realistic manipulator geometry. Fig. 10 shows the number of axes of the reference robot. The axes were assigned as J1, J2, and so on. All axes are in rotation. This Fig. was used for the demonstration purpose of the axes. The actual rotation of each axis can be found in (Fanuc Robotics, 2008). Modelled robot geometry was simplified and designed in Solidworks as shown in Fig. 11 keeping the important features of the reference robot manipulators. Despite simplification to manipulator geometries for physical analysis, motions of each manipulator were identical to the real robot, except the fact that virtual robot would have one DOF i.e. J6 less. The dimensions were based on approximate scale of the reference robot. Small holes were created in each component and used as reference points for assembly into complete robot geometry. These holes would also be useful for defining joints in finite element analysis.

As shown in Fig. 11, the virtual robot has five axes. Each axis would be called J1, J2, and so on just as assigning in the reference robot specification. Solidworks model was saved as the IGES file format that can be exported to finite element package for simulation of physical behaviour.

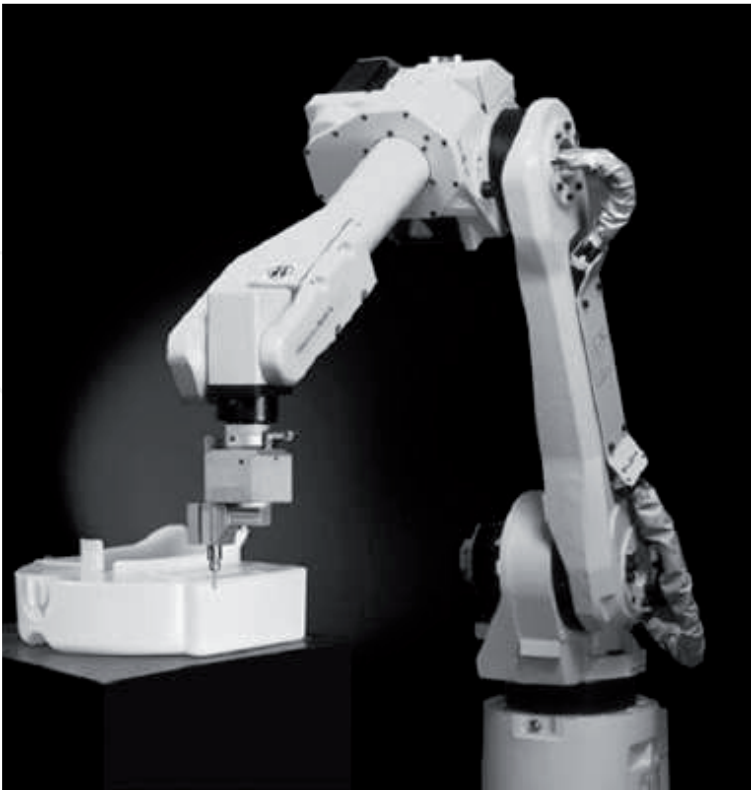


Fig. 9. Reference robot – model FANUC M-6iB (Fanuc Robotics, 2008).

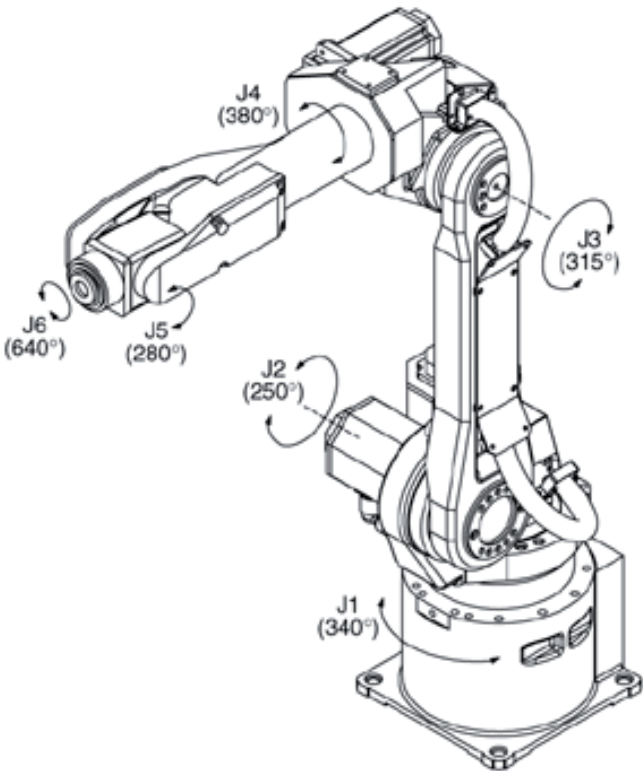


Fig. 10. The axes of the reference robot manipulators (Fanuc Robotics, 2008)

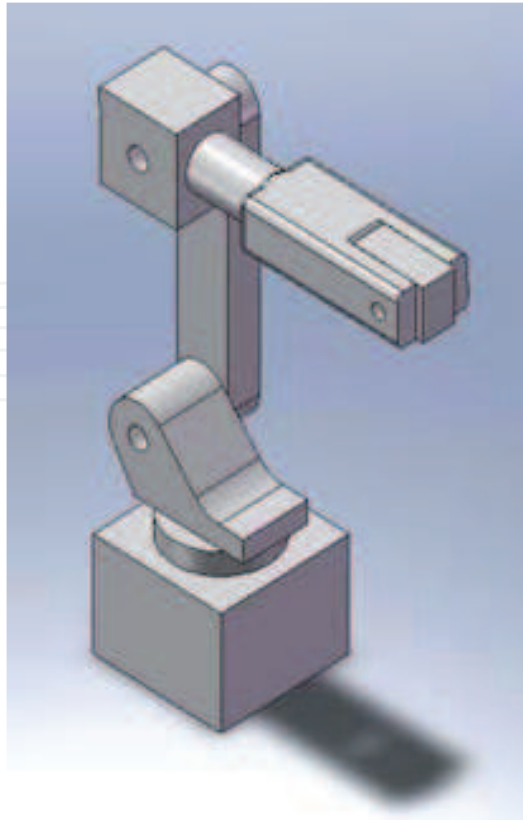


Fig. 11. Simplified robot model geometry

5.1 Finite element analysis

The model geometry of robot manipulators was imported from Solidworks into finite element environment for physical analysis. General-purpose FE codes (ALGOR version 21.2) were tested for the analysis. Fig. 12 shows the discretized model where 8-node brick elements were used as element type for all components. Total number of elements used was 20875 brick elements in manipulators (J1 – J5) and 736 beam elements at joints. According to the available information, steel material (AISI 4130) was used for J5 and the base and the remaining parts were made of aluminum 2024-T3. Linear material model was assumed and the properties of these materials were modelled with data available in FE package material library. Element formulations were solved for large displacement using total Lagrangian approach.

In order to define five axes in the manipulators, joint mesh method and beam elements were used. Pin joint type was chosen for relative movement of each manipulator. Beam elements were selected for joint mesh as these elements can withstand the moment induced. Boundary conditions were then defined at each joint of the manipulator using local coordinate system assigned at joints to create motion being consistent with the reference robot. As this type of robot has to be fixed on the floor, at ceiling or on the table when in use, all nodes at the base were constrained in all directions. To rotate manipulator J1 relative to J2, nodal rotation about Z-axis was prescribed at its joint. At the same time, J2 is rotating about its own axis X as well as swinging about Z. The former DOF is defined by assigning initial velocity at corresponding joint and the latter by using rotational DOF at the node joining with J4. Similarly, the remaining DOFs were defined to create motions of other

manipulators. To eliminate the unknowns in solving FE equations, additional nodal and surface boundary conditions were applied on all joint node and surfaces of manipulators respectively in such a way that translation or rotation in unnecessary directions were constrained.

Mechanical event simulation (MES) with linear material model was used to analyze the kinematics as well as physical response due to dynamic effects. MES can simultaneously analyze mechanical behaviour involving large deformations, nonlinear response, kinematic motion and forces caused by that motion and predict resulting stresses and strains. Details can be found in (ALGOR, 2008).

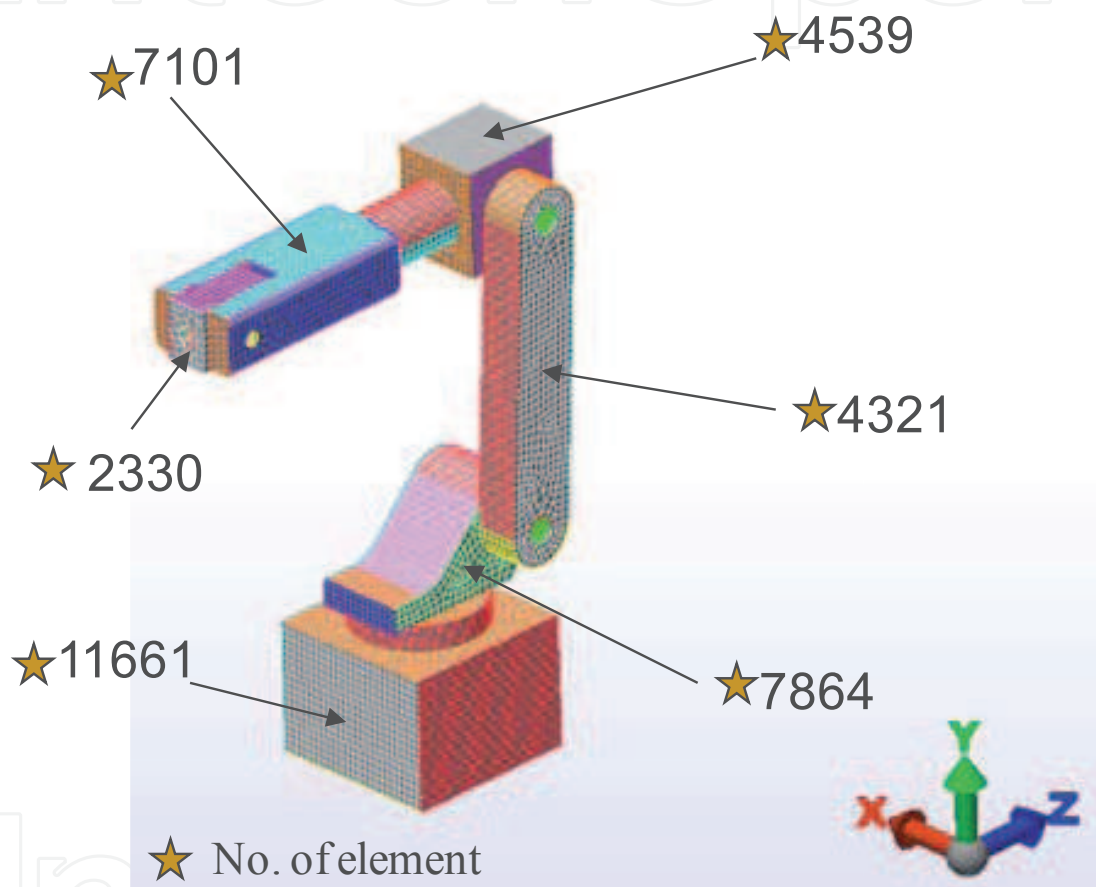


Fig. 12. Finite element model of robot manipulators.

5.2 Discussion

The idea here is the creation of multi-axis positioning with consideration of physical behaviour through FEM. This was done by manipulating available FE codes constructed for boundary conditions. Furthermore, a lot of information can be extracted from the FE analysis. The useful information on dynamic structural behaviour includes stress, strain, displacements and moment/torque. The stress-strain and displacement plots, for instance, are very important to evaluate the design. The information on torque is useful to choose a proper motor size for the real robot. Inertia effect, weight effect and can also be taken into account in the model. In this simple example, these factors were ignored.

Motions of manipulators were analyzed only for 0.1 s due to computational limitation of FE code applied although the motions of the manipulators in reference robot were specified for 1s in the specification. Even that the computing time took 16 hrs to complete the analysis. Fig. 13 demonstrates the positioning of robot manipulators (axes) as defined by boundary conditions. It also depicts the deformation of each manipulator due to dynamic loads. Fig. 14 – 18 shows the predicted displacements of corresponding manipulators. Table 1 shows comparison between the rotation of the reference robot manipulators and that of virtual robot manipulators. The computational results are consistent with the motion of the reference robot manipulators from view point of motion speed.

Axes	Ref robot spec.		Computed rotation	% error
	1 s	0.18 s		
J1	150°	27°	26.5°	2
J2	160	28.8	28.87	0.02
J3	170	30.6	30.8	0.06
J4	400	72	72	0
J5	400	72	71.93	0

Table 1. Comparison between the actual and computed rotations

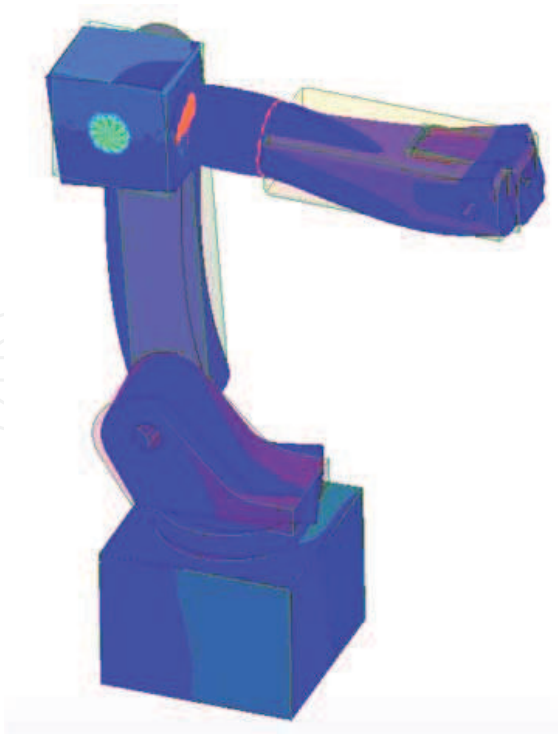


Fig. 13. The deformed plot of robot manipulators.

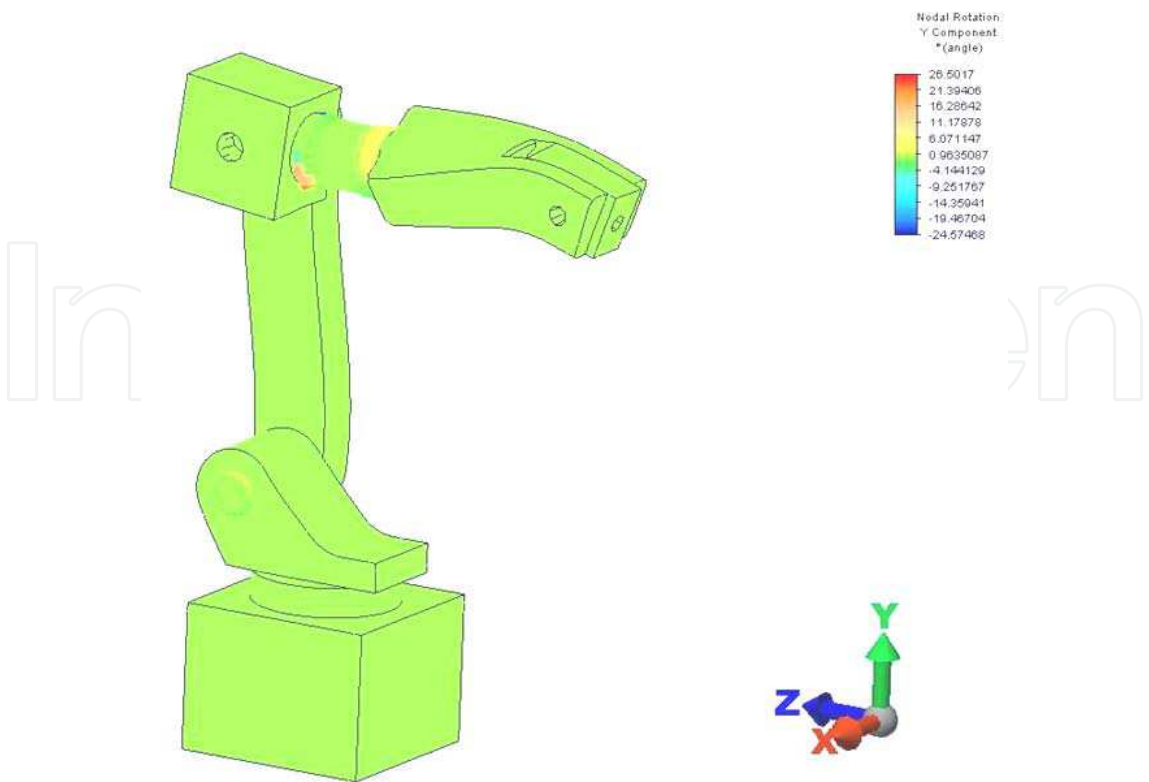


Fig. 14. Computed Rotational displacement of J1

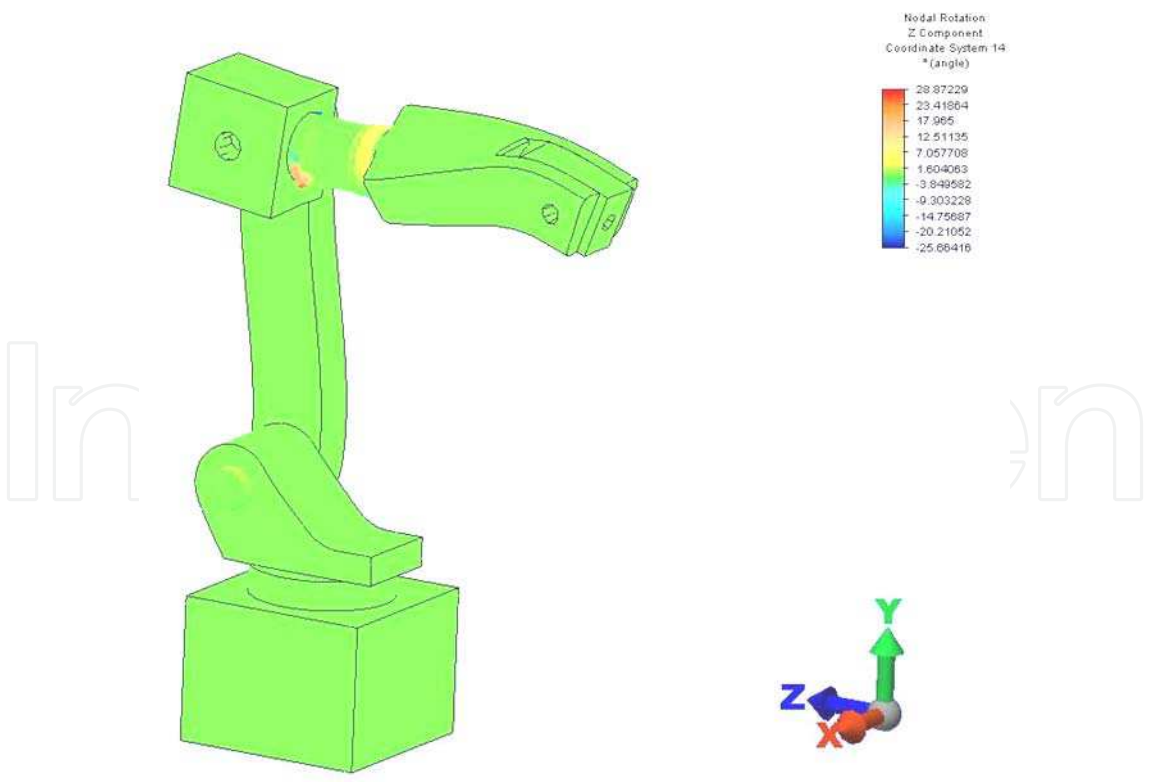


Fig. 15. Computed Rotational displacement of J2

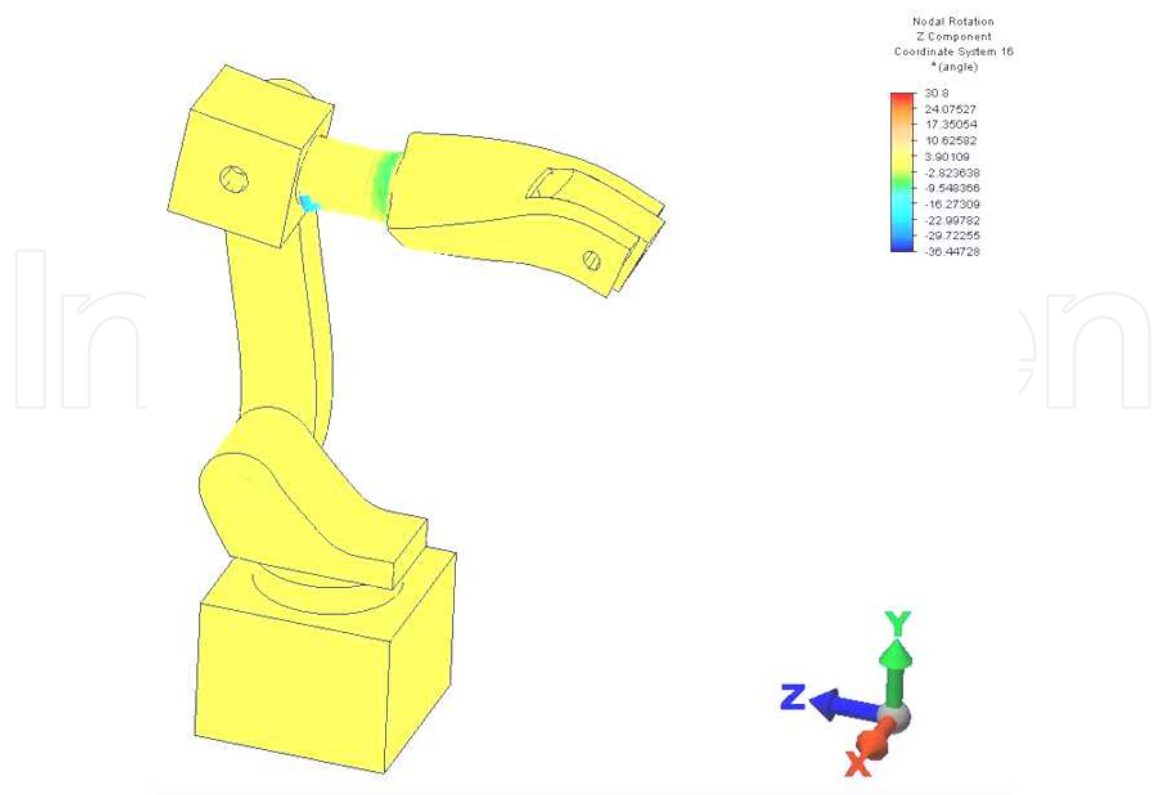


Fig. 16. Computed Rotational displacement of J3

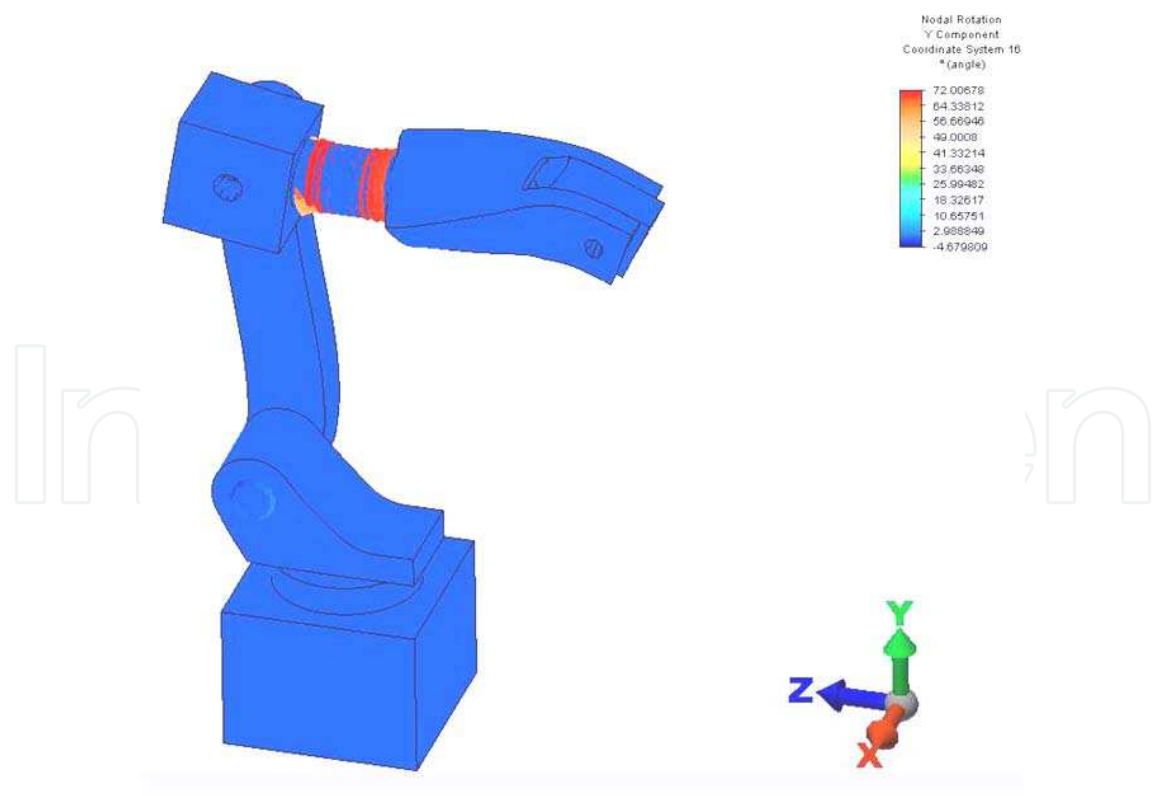


Fig. 17. Computed Rotational displacement of J4

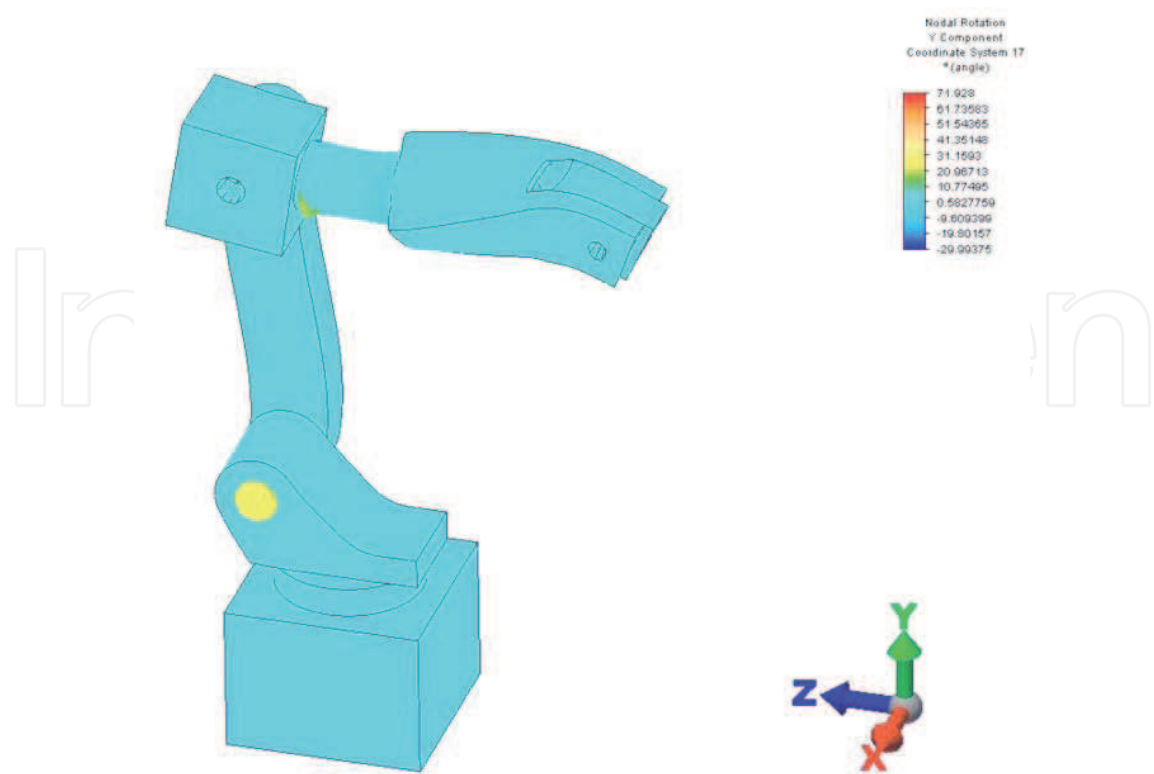


Fig. 18. Computed Rotational displacement of J5

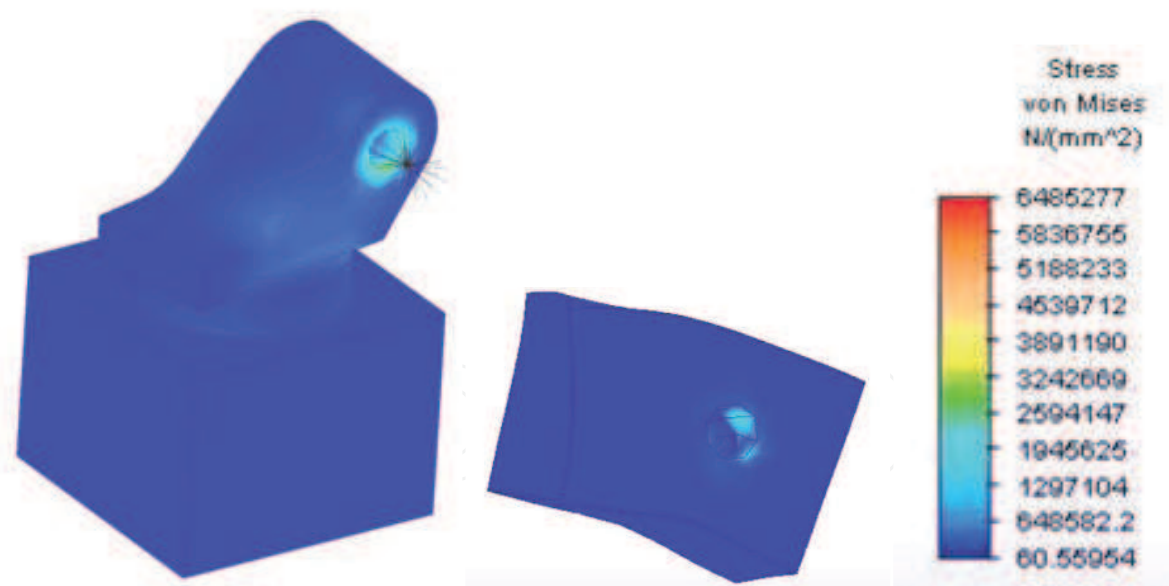


Fig. 19. Computed effective stress.

Fig. 19 shows contour plots of dynamic stress at joints. Contour plots show the deformation of each component clearly. The twisting and bending of the manipulator can be clearly seen. The main reason is that in current model the friction effect at the contact and clearance were ignored. The sizes of holes in mating parts for assembly were the same as if zero clearance. Presence of bearing was not there. The size and geometry of manipulators were merely

approximation which was not based on proper design calculations. Consequently, severe twisting would occur especially at the base where it supports all manipulators and endure the dynamically-exerted forces. The computed effective stress is useful to determine the failure of each manipulator due to dynamic loading. Fig. 20 shows the computed maximum torque required to rotate J5. Computed torque can be used to estimate motor power required for robot manipulators.

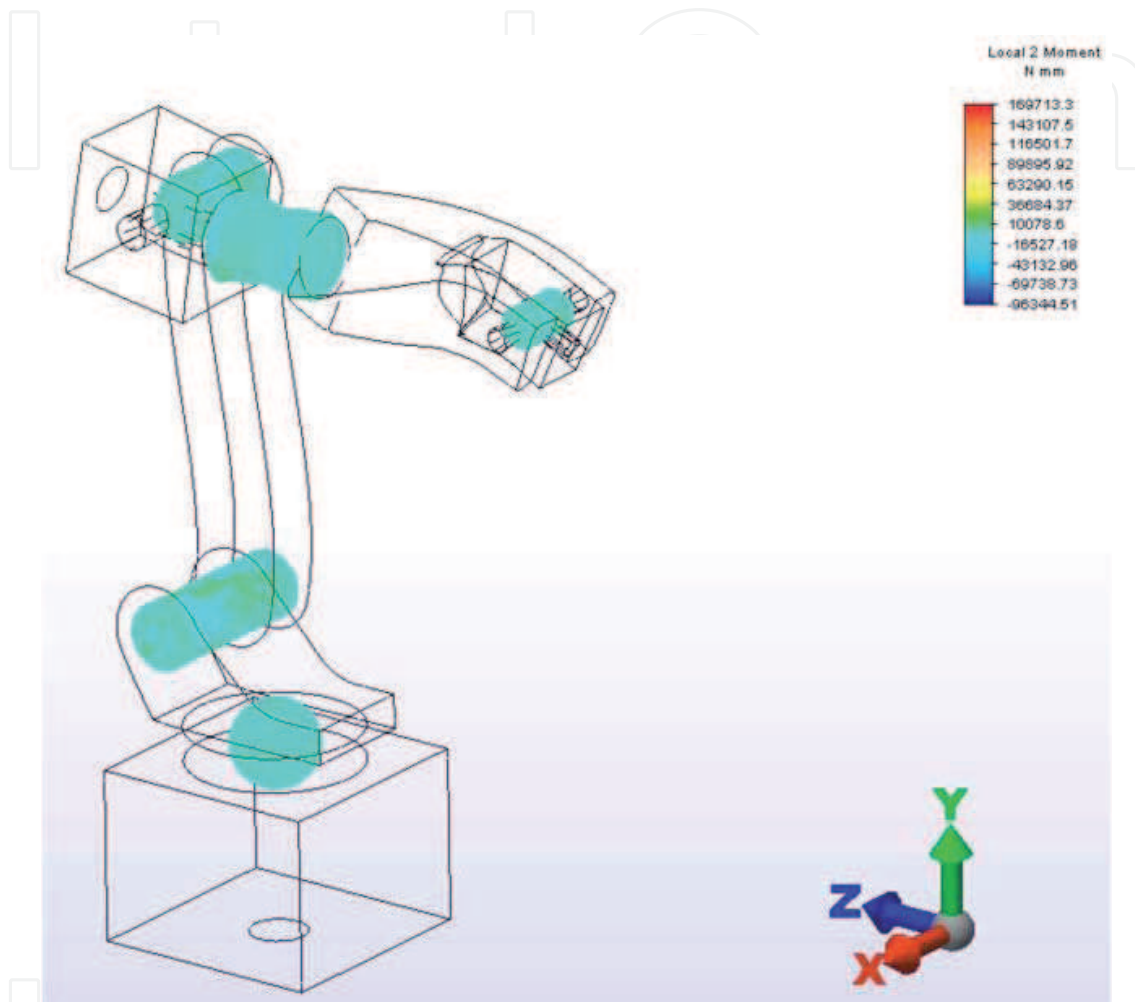


Fig. 20. Computed torque.

6. Conclusion

The FEM is capable of analyzing multi-axes positioning as well as physical response to dynamic load. The current FE code used for dynamic analysis is flexible to analyze physical response of the system due to multi-axes positioning involving large displacement.

Capability of FE code may be different from that of another i.e. FE analysis greatly depends on the FE code. Some FE codes have limitations. Such limitations are usually model size (i.e. numbers of elements and nodes), meshing, material models, analysis type and solver type. In addition, the hardware capacity affects computational power. In future work, weight factor, mesh design and an appropriate material model will be taken into account in the finite element modeling and analysis.

7. References

- ALGOR, *User's Guide*, 2008.
- Angeles, J. (2003). *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, Second Edition, Springer-Verlag New York, Inc.
- Billingsley, J. (1985). *Robots and automated manufacture*, IEEE Control Engineering Series 28, Peter Peregrinus Ltd, London, pp. 11-23, 29-33.
- Chandrupatla, T.R. & A.D. Belegundu, A.D. (2002). *Introduction to Finite Elements in Engineering*, Third Edition, Prentice Hall, ISBN : 0-13-207036-7.
- Cleery, B.M. & Mathur, N. (2008). Mechanical Engineering Magazine, ASME 130 (6) June 2008, pp.30-33.
- Fanuc Robotics (2008). <http://www.robots.com/robots.php>.
- Hutton, D.V. (2004). *Fundamentals of Finite Element Analysis*, International Edition, McGraw Hill (Asia), Singapore, ISBN : 007-124160-4
- Keramas, J. G. (1999). *Robot Technology Fundamentals*, Delmar Publishers, pp.24-25.
- Lau, H. Y. K. ; Mak, K. L. & Lu, M. T. H. (2003). A Virtual Design Platform for Interactive Product Design and Visualization, *Journal of Materials Processing Technology* 139, 2003, pp. 402-407.
- Mackerle, J. (1999). Finite Element Analysis and Simulation of Machining. *Journal of Materials Processing Technology* 86: 17-44.
- Manseur, R. (2006). *Robot Modeling and Kinematics*, Da Vinci Engineering Press, Massachusetts, ISBN : 1-58450-851-5.
- Rahman, M. (2004). Modeling and Measurement of Multi-axis Machine Tools to Improve Positioning Accuracy in a Software Way, Academic Dissertation presented with the assent of the Faculty of Technology, University of Oulu, for public discussion in Kuusamonsali (Auditorium YB210), Linnanmaa, June 4th, 2004.
- Zlajpah, L. (2008). Simulation in Robotics, *Mathematics and Computers in Simulation* (2008), Available online at www.sciencedirect.com.



Advances in Robot Manipulators

Edited by Ernest Hall

ISBN 978-953-307-070-4

Hard cover, 678 pages

Publisher InTech

Published online 01, April, 2010

Published in print edition April, 2010

The purpose of this volume is to encourage and inspire the continual invention of robot manipulators for science and the good of humanity. The concepts of artificial intelligence combined with the engineering and technology of feedback control, have great potential for new, useful and exciting machines. The concept of eclecticism for the design, development, simulation and implementation of a real time controller for an intelligent, vision guided robots is now being explored. The dream of an eclectic perceptual, creative controller that can select its own tasks and perform autonomous operations with reliability and dependability is starting to evolve. We have not yet reached this stage but a careful study of the contents will start one on the exciting journey that could lead to many inventions and successful solutions.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

T.T. Mon, F.R. Mohd Romlay and M.N. Tamin (2010). Role of Finite Element Analysis in Designing Multi-axes Positioning for Robotic Manipulators, *Advances in Robot Manipulators*, Ernest Hall (Ed.), ISBN: 978-953-307-070-4, InTech, Available from: <http://www.intechopen.com/books/advances-in-robot-manipulators/role-of-finite-element-analysis-in-designing-multi-axes-positioning-for-robotic-manipulators>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen